COL-724 : A3: Programming Assignment 3: SDN and Ryu

Advanced Computer Networks

COL724

A3

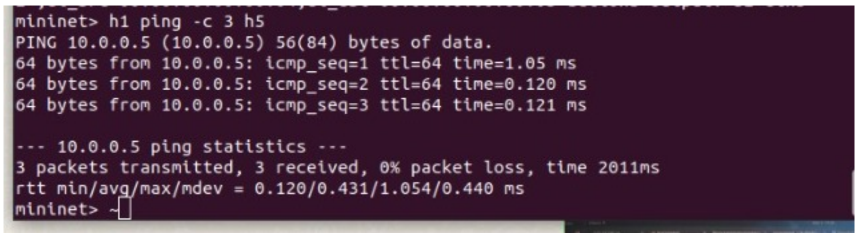
Samyak Jain: 2020CS50667

Written under guidance of ***Prof. Tarun Mangla***

Keywords: SDN,Ryu,iperf, ping, dpctl

1 Hub Controller and Learning Switch

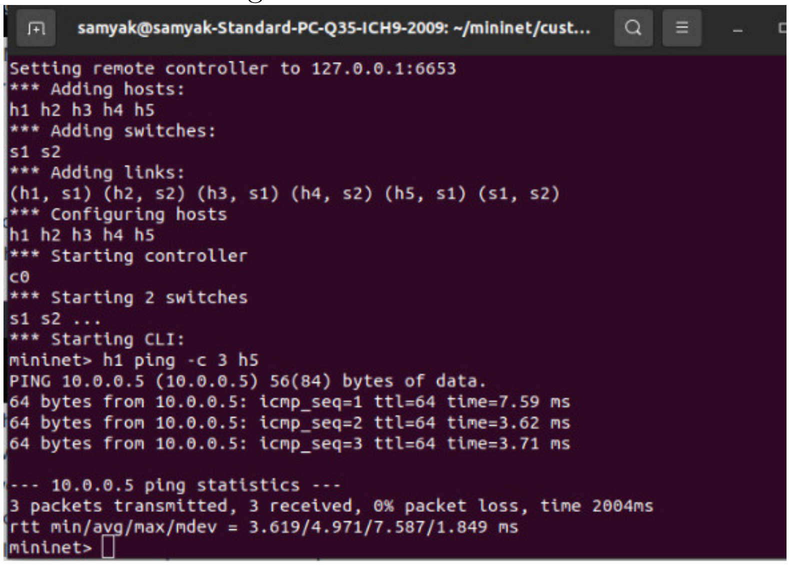
1.1 part A



```
mininet> h1 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.120 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.121 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.120/0.431/1.054/0.440 ms
mininet> ~
```

This is for learning switch

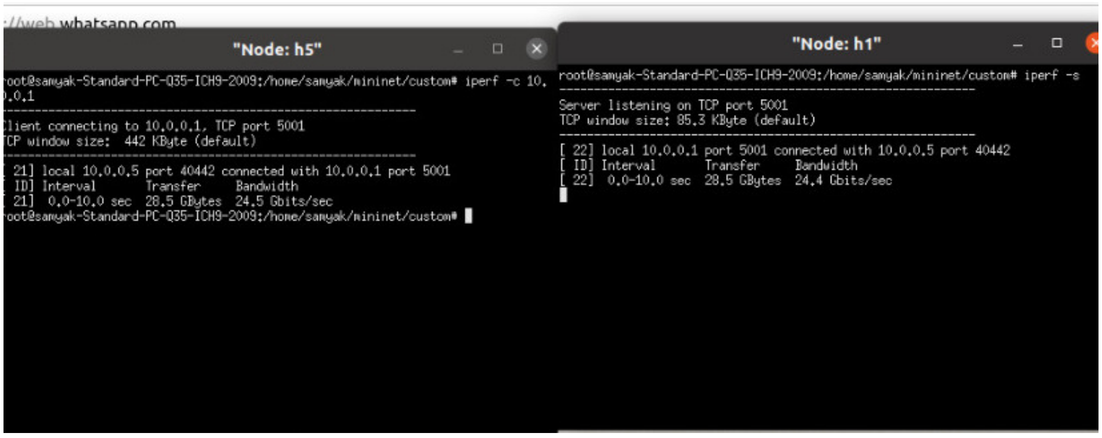


```
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (h3, s1) (h4, s2) (h5, s1) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> h1 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=7.59 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=3.62 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=3.71 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.619/4.971/7.587/1.849 ms
mininet> 
```

This is for controller hub

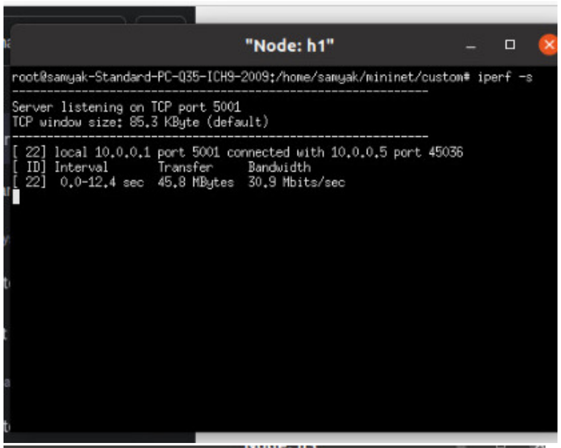
1.2 part B



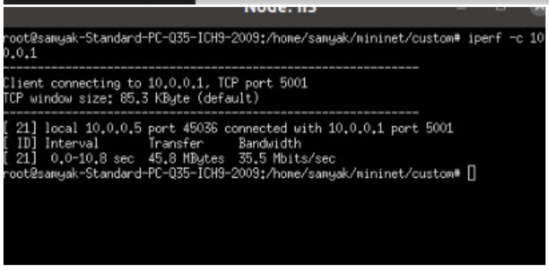
```
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# iperf -c 10.
).0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size:  442 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.5 port 40442 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0-10.0 sec  28.5 GBytes  24.5 Gbits/sec
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# 
```

```
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 22] local 10.0.0.1 port 5001 connected with 10.0.0.5 port 40442
[ ID] Interval       Transfer     Bandwidth
[ 22]  0.0-10.0 sec  28.5 GBytes  24.4 Gbits/sec
```

This is for learning switch



```
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 22] local 10.0.0.1 port 5001 connected with 10.0.0.5 port 45036
[ ID] Interval       Transfer     Bandwidth
[ 22]  0.0-12.4 sec  45.8 MBytes  30.9 Mbits/sec
```



```
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# iperf -c 10
0.0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.5 port 45036 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0-10.8 sec  45.8 MBytes  35.5 Mbits/sec
root@samyak-Standard-PC-Q35-ICH9-2009:/home/samyak/mininet/custom# []
```

This is for controller hub

1.3 part C

```
 *** s1 ------------------------------------------------------------------------
cookie=0x0, duration=74.904s, table=0, n_packets=686896, n_bytes=30192288472, in_por
eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=74.902s, table=0, n_packets=580032, n_bytes=38282248, in_port="
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
cookie=0x0, duration=9.855s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=9.850s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
cookie=0x0, duration=9.829s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=9.827s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
cookie=0x0, duration=9.813s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=9.811s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
cookie=0x0, duration=9.777s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
cookie=0x0, duration=9.776s, table=0, n_packets=2, n_bytes=140, in_port="s1-
```

```
 eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=9.768s, table=0, n_packets=3, n_bytes=238, in_port="s1-
 eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=9.765s, table=0, n_packets=2, n_bytes=140, in_port="s1-
 eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=9.744s, table=0, n_packets=3, n_bytes=238, in_port="s1-
 eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=9.738s, table=0, n_packets=2, n_bytes=140, in_port="s1-
 eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=9.701s, table=0, n_packets=3, n_bytes=238, in_port="s1-
 eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=9.695s, table=0, n_packets=2, n_bytes=140, in_port="s1-
 eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=9.667s, table=0, n_packets=3, n_bytes=238, in_port="s1-
 eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=9.663s, table=0, n_packets=2, n_bytes=140, in_port="s1-
 eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=74.918s, table=0, n_packets=686896, n_bytes=30192288472, in_port
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=74.914s, table=0, n_packets=580032, n_bytes=38282248, in_port="
 eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=9.827s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=9.822s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=9.782s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=9.770s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=9.760s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=9.748s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=9.721s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=9.706s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=9.689s, table=0, n_packets=3, n_bytes=238, in_port="s2-
```

```
eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
cookie=0x0, duration=9.671s, table=0, n_packets=2, n_bytes=140, in_port="s2-
eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
cookie=0x0, duration=9.643s, table=0, n_packets=3, n_bytes=238, in_port="s2-
eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
cookie=0x0, duration=9.639s, table=0, n_packets=2, n_bytes=140, in_port="s2-
eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2
```

For the case of controller, following are the flows:

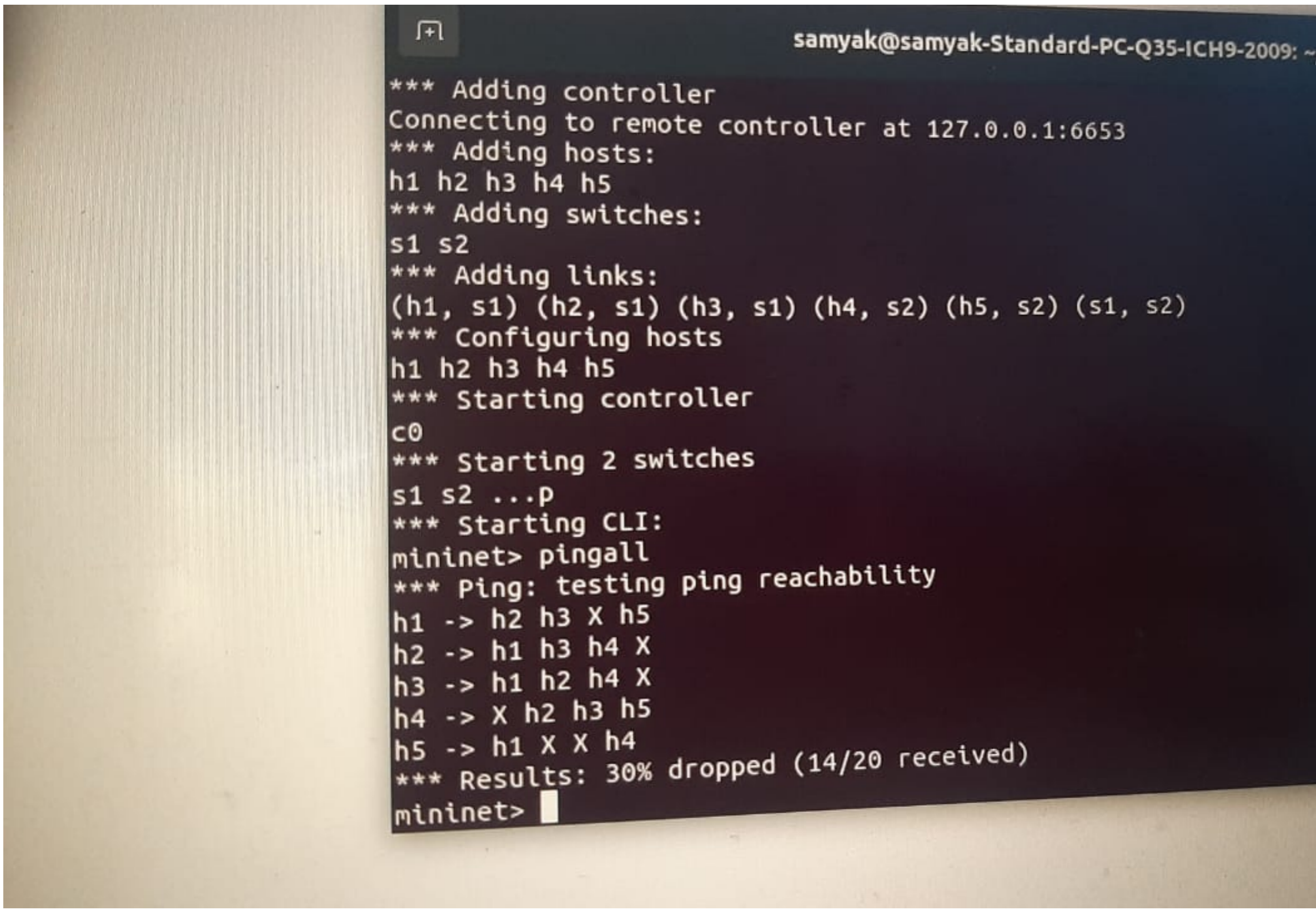
```
  *** s1 ------------------------------------------------------------------
 cookie=0x0, duration=109.966s, table=0, n_packets=32294, n_bytes=32566347, priority=
 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------
 cookie=0x0, duration=109.999s, table=0, n_packets=20843, n_bytes=31810577, priority=
 actions=CONTROLLER:65535
```

2 Firewall and Monitor

```
  *** s1 ------------------------------------------------------------------
cookie=0x0, duration=314.523s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=314.521s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
cookie=0x0, duration=314.511s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=314.509s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
cookie=0x0, duration=314.499s, table=0, n_packets=4, n_bytes=224, in_port="s1-
eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=304.497s, table=0, n_packets=5, n_bytes=322, in_port="s1-
eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
cookie=0x0, duration=304.496s, table=0, n_packets=4, n_bytes=224, in_port="s1-
eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
cookie=0x0, duration=304.486s, table=0, n_packets=3, n_bytes=238, in_port="s1-
eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
cookie=0x0, duration=304.484s, table=0, n_packets=2, n_bytes=140, in_port="s1-
eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
```

```
 cookie=0x0, duration=304.475s, table=0, n_packets=4, n_bytes=280, in_port="s1-
 eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=304.472s, table=0, n_packets=3, n_bytes=182, in_port="s1-
 eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=304.430s, table=0, n_packets=4, n_bytes=224, in_port="s1-
 eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=294.452s, table=0, n_packets=3, n_bytes=238, in_port="s1-
 eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=294.450s, table=0, n_packets=2, n_bytes=140, in_port="s1-
 eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=294.438s, table=0, n_packets=4, n_bytes=224, in_port="s1-
 eth4",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=314.505s, table=0, n_packets=4, n_bytes=224, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=304.503s, table=0, n_packets=5, n_bytes=322, in_port="s2-
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=304.499s, table=0, n_packets=4, n_bytes=224, in_port="s2-
 eth3",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=304.481s, table=0, n_packets=4, n_bytes=280, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=304.475s, table=0, n_packets=3, n_bytes=182, in_port="s2-
 eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=304.466s, table=0, n_packets=4, n_bytes=224, in_port="s2-
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=294.458s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=294.451s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=294.443s, table=0, n_packets=4, n_bytes=224, in_port="s2-
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=274.431s, table=0, n_packets=3, n_bytes=238, in_port="s2-
 eth2",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth1"
 cookie=0x0, duration=274.426s, table=0, n_packets=2, n_bytes=140, in_port="s2-
 eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
```

following is he screenshot of the firewall on pingall. X clearly show that the communication between restricted hosts could not be done



```
samyak@samyak-Standard-PC-Q35-ICH9-2009: ~
*** Adding controller
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...p
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X h5
h2 -> h1 h3 h4 X
h3 -> h1 h2 h4 X
h4 -> X h2 h3 h5
h5 -> h1 X X h4
*** Results: 30% dropped (14/20 received)
mininet>
```

The most important way to minimize rules is by using patterns present in the header. We can use the prefixes in the IP and match using that instead of enumerating a different rule for all possible IPs. This is because there might be patters in the packet communication for particular sets of IPs. These are generaly exploited.
Grouping and Hierarchical Rules: Organize your firewall rules into groups and hierarchies. Create a set of high-level rules that apply to broad categories of traffic and only add detailed rules when necessary. This helps in reducing the number of rules that need to be checked for each packet.
Consolidation: Combine similar rules into a single rule whenever possible. For example, if you have multiple rules that block traffic from different source IPs but to the same destination IP, you can consolidate them into a single rule to save resources and reduce complexity.

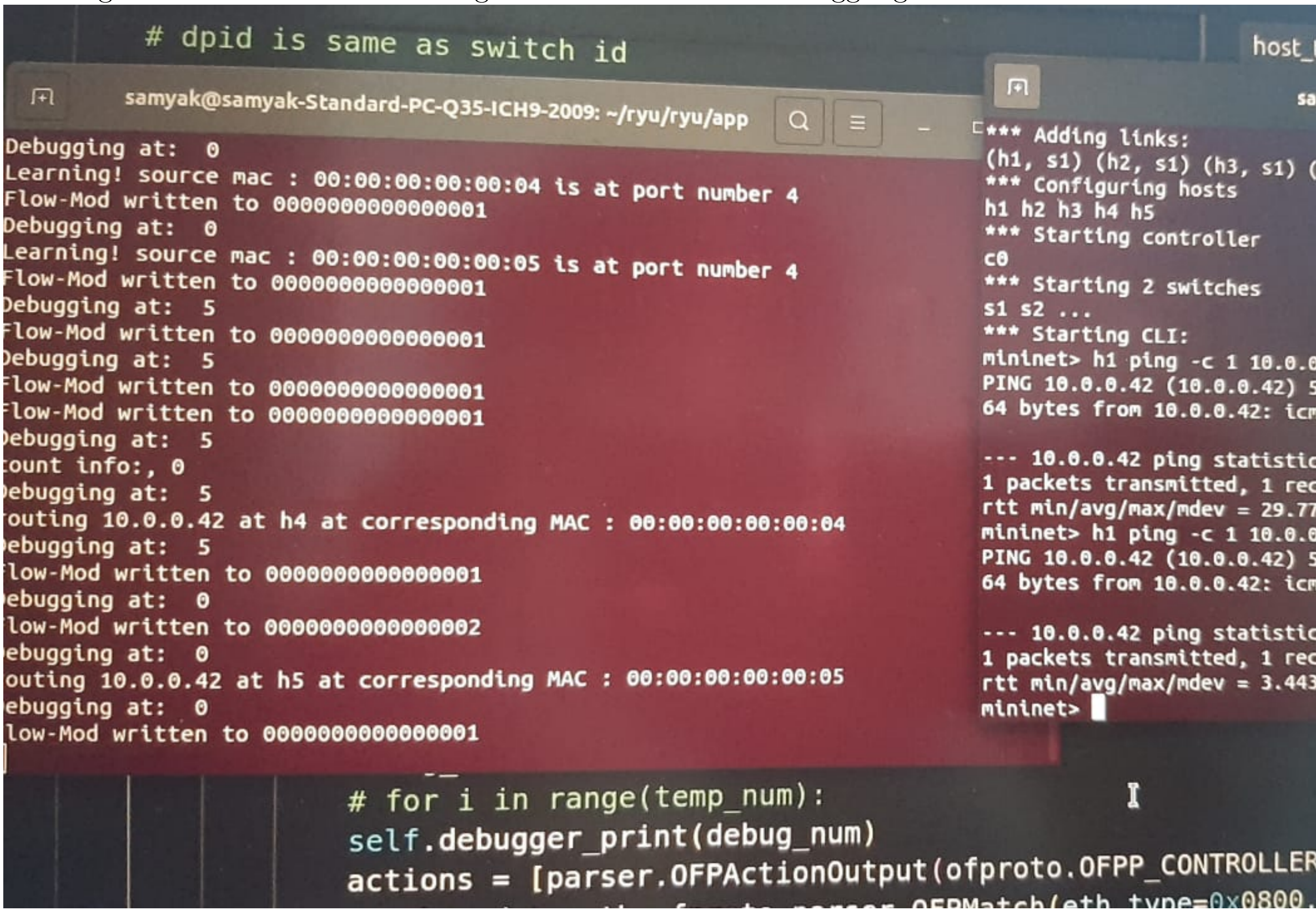
2.0.1 part c

To implement rules in real time which do not interfere, it is critical to define some order in which more critical rules are checked first and later the less precedence ones. We must insert the rules in real time in this order to prevent interference with other rules.
Time-Based Rules: Implement time-based rules. Firewall policies can be configured to

apply only during specific time periods, such as business hours or maintenance windows. This allows the operator to apply new rules during off-peak times when minimal interference is expected.
Implement Incremental Changes: Rather than making broad changes all at once, implement firewall policies incrementally. This allows the operator to assess the impact of each change before proceeding to the next one.

2.1 Load Balancer

To implement this part, I have used a variable called count which is maintaining which erver was last used and I take it mod 2 and increment it everytime. This acts as a toggle selecting the correct server. Following is the screenshot of the toggling:



```
# dpid is same as switch id
```

samyak@samyak-Standard-PC-Q35-ICH9-2009: ~/ryu/ryu/app

```
Debugging at:  0
Learning! source mac : 00:00:00:00:00:04 is at port number 4
Flow-Mod written to 0000000000000001
Debugging at:  0
Learning! source mac : 00:00:00:00:00:05 is at port number 4
Flow-Mod written to 0000000000000001
Debugging at:  5
Flow-Mod written to 0000000000000001
Debugging at:  5
Flow-Mod written to 0000000000000001
Flow-Mod written to 0000000000000001
Debugging at:  5
Count info:, 0
Debugging at:  5
Routing 10.0.0.42 at h4 at corresponding MAC : 00:00:00:00:00:04
Debugging at:  5
Flow-Mod written to 0000000000000001
Debugging at:  0
Flow-Mod written to 0000000000000002
Debugging at:  0
Routing 10.0.0.42 at h5 at corresponding MAC : 00:00:00:00:00:05
Debugging at:  0
Flow-Mod written to 0000000000000001
```

```
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> h1 ping -c 1 10.0.0
PING 10.0.0.42 (10.0.0.42) 5
64 bytes from 10.0.0.42: icm

--- 10.0.0.42 ping statistic
1 packets transmitted, 1 rec
rtt min/avg/max/mdev = 29.77
mininet> h1 ping -c 1 10.0.0
PING 10.0.0.42 (10.0.0.42) 5
64 bytes from 10.0.0.42: icm

--- 10.0.0.42 ping statistic
1 packets transmitted, 1 rec
rtt min/avg/max/mdev = 3.443
mininet>
```

```
# for i in range(temp_num):
self.debugger_print(debug_num)
actions = [parser.OFPActionOutput(ofproto.OFPP_CONTROLLER
```

Following are the learnt rules:

```
   cookie=0x0, duration=17.624s, table=0, n_packets=0, n_bytes=0,
   priority=2,ip,nw_src=10.0.0.1,nw_dst=10.0.0.42 actions=CONTROLLER:65535
cookie=0x0, duration=17.659s, table=0, n_packets=2, n_bytes=140, priority=1,in_port=
eth4",dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
```

```
 cookie=0x0, duration=17.618s, table=0, n_packets=1, n_bytes=42, priority=1,in_port="
 eth1",dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=11.251s, table=0, n_packets=2, n_bytes=140, priority=1,in_port=
 eth4",dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=11.236s, table=0, n_packets=1, n_bytes=42, priority=1,in_port="
 eth2",dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=6.753s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="
 eth4",dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=6.743s, table=0, n_packets=1, n_bytes=42, priority=1,in_port="s
 eth3",dl_dst=00:00:00:00:00:05 actions=output:"s1-eth4"
 cookie=0x0, duration=33.787s, table=0, n_packets=52, n_bytes=5558, priority=0
 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=17.714s, table=0, n_packets=0, n_bytes=0, priority=1,in_port="s
 eth1",dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=17.714s, table=0, n_packets=2, n_bytes=140, priority=1,in_port=
 eth2",dl_dst=00:00:00:00:00:01 actions=output:"s2-eth3"
 cookie=0x0, duration=17.649s, table=0, n_packets=5, n_bytes=322, priority=1,in_port=
 eth3",dl_dst=00:00:00:00:00:05 actions=output:"s2-eth2"
 cookie=0x0, duration=11.309s, table=0, n_packets=2, n_bytes=140, priority=1,in_port=
 eth2",dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=11.308s, table=0, n_packets=0, n_bytes=0, priority=1,in_port="s
 eth1",dl_dst=00:00:00:00:00:02 actions=output:"s2-eth3"
 cookie=0x0, duration=6.810s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="
 eth2",dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=6.809s, table=0, n_packets=0, n_bytes=0, priority=1,in_port="s2
 eth1",dl_dst=00:00:00:00:00:03 actions=output:"s2-eth3"
 cookie=0x0, duration=33.818s, table=0, n_packets=46, n_bytes=5062, priority=0
 actions=CONTROLLER:65535
```

2.1.1 New load balancer

For this type of load balancer we would need a feedback function and resource function. Resource function would tell the feedback function how many resources are available and feebac function will report this to the controller which will then use some "scheduling" policy to optimally determine which server is the best torespond to the next packet coming from the host h1 h2 h3. This will take into account the real time load geenrated by packets of h2 h2 h3. In our present scenario, we do round robin which is often not considered fair because if h1 continuously sends large packet while h2 and h3 send small packets, then h1 is able to get more benefit as it is sending larger packets while also getting same number of chances as h2 and h3 . Thus it is not fair.